Problem 1 (0 Points)
Read the solution to Assessment 1 and write a C/C++ program that demonstrates Problems 4, 5 and 6 on assessment 1.

Problem 2 (6 Points)
Write a C/C++ program that will eventually evolve into a special command-line interpreter. For now, the program will just do some simple lexical analysis and outputting. In the description below, a command-line is a sequence of characters separated by one or more consecutive whitespace characters. A white-space character is either a space or a tab character. Each non-white-space character sequence is called a word.

- The program has the following synopsis:
  
  \texttt{hw1-2 [-v] < Commands}

  The \texttt{-v} flag means to run in verbose mode (described below) and is an optional flag. The default is to run in normal mode. Normally, command lines (see below) for a special language are read from \texttt{stdin}. In this assignment, we are only interested in the syntax of the language and the semantics of the language is left undefined.

- Your program should do the following for each command line:

  - Display on \texttt{stdout} the command line exactly as it appears on input.
  - Place the words into a 100-byte buffer in \textit{internal format}. Suppose that there are \( N \) words in the command. The buffer contains \( N + 1 \) pointers immediately followed by the character strings (NUL-terminated) packed one string after the other. The first \( N \) words are pointers to the words in the buffer. Word \( N \) (counting from 0) contains a null pointer indicating the end of the pointer array.
  - Then, display the contents of the buffer if verbose mode has been selected. In this format, pointers are displayed in character is displayed in hexadecimal (field is separated by a space character. For example, \texttt{'set foo 32'} might be displayed as follows:

    \begin{verbatim}
    0xbfffe7fc 0xbfffe800 0xbfffe804 (nil) 73 65 74 0 66 6f 6f 0 33 32 0
    \end{verbatim}

  - Your program should exit when either the first word is \texttt{exit} or an \textit{end-of-file} condition is encountered on \texttt{stdin}.

Here is a detailed description of the language:

- Each command is separated by a newline character (\texttt{"\n}).
- All commands have a variable number of arguments which are separated from each other by white-space. However, you can assume that a command never has more than five words.
Submit:

- Source listing
- Test output (with and without '-v')
- A short explanation of why the test output indicates that your program is operating correctly.

The following man pages might be useful: `strcmp(3)`, `memset(3)`, `isdigit(3)`, `atoi(3)`, `strtol(3)`, `strtok(3)`, `printf(3)`, `exit(3)`, `memcpy(3)`, `fgets(3)`, `strlen(3)` and `ascii(7)`. The digit in parentheses indicates the probable section where you can find the man page; i.e., 'man 3 memset' looks for `memset` in section 3. Note that in Solaris, you will need the `-s` flag; i.e., 'man -s 3 memset'.